Title	2NPDAによるシミュレーションと未解決問題(計算アルゴリズムと計算量の基礎理論)
Author(s)	岩田, 茂樹; 笠井, 琢美
Citation	数理解析研究所講究録 (1989), 695: 27-36
Issue Date	1989-06
URL	http://hdl.handle.net/2433/101407
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

数理解析研究所講究録
695巻 1989年 27-36

２ＮＰＤＡによるシミュレーションと未解決問題

東海大学情報処理研究教育施設　岩田茂樹（Shigeki Iwata）
電気通信大学 情報工学科　笠井琢美（Takumi Kasai）

あらまし　非決定性チューリング機械やオルタネーティングチューリング機械による計算のある種のクラスを2方向非決定性プッシュダウンオートマタでシミュレートすることについて述べる．NSPACE(S(n))（ASPACE(S(n))）を S(n) 領域限定の非決定性（オルタネーティング，それぞれ）チューリング機械で受理される言語のクラスとする．また 2NPDA（$2NPDA_{poly}$）を（多項式時間限定の，それぞれ）2方向非決定性プッシュダウンオートマタで受理される言語クラスとする．すると定数 c>0 に対して

(1) $NSPACE(c \log n) \subseteq 2NPDA_{poly}$,
(2) $ASPACE(c \log n) \subseteq 2NPDA$

が得られた．形式言語理論の未解決問題にこれらの結果を応用することについても述べる．

1．はじめに

2方向プッシュダウンオートマトンは Gray, Harrison and Ibarra [4] により導入され，以来多くの研究がなされてきた．

Aho, Hopcroft and Ullman [1] は2方向決定性，非決定性プッシュダウンオートマタをそれぞれチューリング機械でシミュレートした．Cook [2] は2方向決定性，非決定性多ヘッドプッシュダウンオートマタで受理される言語のクラスはいずれも多項式時間限定チューリング機械で受理される言語のクラスと同じであることを示した．Galil [3] は入力を入力長回くり返しつなげることを利用して，決定性2方向プッシュダウンオートマタにより受理される言語のクラスと対数領域限定決定性チューリング機械により受理される言語のクラスは異なることを示した．プッシュダウンオートマタと対数領域限定チューリング機械に関する研究が Sudborough [7,8,9] によってなされている．

本報告では定数 c>0 に対し，c log n 領域限定非決定性チューリング機械の計算を多項式時間限定2方向プッシュダウンオートマタでシミュレートすることと，c log n 領域限定オルタネーティングチューリング機械の計算を2方向プッシュダウンオートマタでシミュレートすることを示す．

非決定性とオルタネーティングチューリング機械，並びに2方向非決定性プッシュダウン

オートマタ（2npda と略す）について読者は熟知しているものと仮定する．これらの機械への入力は左右のエンドマーカが付いているとする．第1図に示すように，左エンドマーカ $ にヘッドがあるなら，ヘッドの位置は0番目にあるということにする．機械が計算を始めるときと入力を受理するときは入力ヘッドは0番目の位置にあるとする．この仮定は一般性を失わないことに注意する．

スタック記号の集合をΓとする．プッシュダウンスタックの内容が$\alpha \in \Gamma^*$であるとは，αの左端の記号がスタックの底にあり右端の記号がスタックの一番上にあることとする．NSPACE(S(n))（ASPACE(S(n)) を S(n) 領域限定の非決定性（オルタネーティング，それぞれ）チューリング機械で受理される言語のクラスとし，$NL = \bigcup_c NSPACE(c \log n)$，P を多項式時間限定チューリング機械で受理される言語のクラスとする． 2NPDA = {L| ある 2npda A に対し L=L(A)}，$2NPDA_{poly}$={L| ある多項式時間限定 2npda A に対し L=L(A)} とする．$2DPDA_{poly}$も2方向決定性プッシュダウンオートマタ（2dpda と略す）について同様に定義する．LOGCFL（LOGDCFL）を（決定性,それぞれ）文脈自由言語に対数領域還元可能な言語のクラスとする．LOGCFL についてはいろいろな特徴ずけがなされている．
[6,8,9]

上に述べたクラスの包含関係で現在知られているものを第2図に示す．図中，上から下に線があるのは上のクラスが下のクラスを真に含むことを示し，破線は真であることがまだわかっていないことを示す．図中，線のないクラス間の包含関係はまだわかっていない．

以後，対数の底は 2，記号列 w に対し |w| は w の長さを表すものとする．

2．シミュレーションと結果

この節ではチューリング機械の 2npda による2種類のシミュレーションを示す．始めに 2dpda の基本的性質を示す．

命題1　A を入力長 n の 2dpda とする．計算の途中でプッシュダウン記憶の内容が$\gamma \# \alpha, \gamma, \alpha \in \{0,1\}^*$, $|\alpha| \leq \log n$ であり，ヘッドの位置が左端にあったとする．すると (1) A はαをポップアップした後にヘッドの位置を入力テープ上のα^R番目に移動することができる．ただしα^Rは2進数とする．(2) この後，A はプッシュダウン記憶の内容を再び$\gamma \# \alpha$にし，入力ヘッドの位置を左端の記号上に移動することができる．

証明　@を使用していない A のスタック記号とする．命題の前半を証明するには，αをスタックからポップアップし入力ヘッドをα^R番目の位置に移動する A の手続きを示せばよい．

```
while the symbol on the stack top is not # do
   while input head is not on the leftmost symbol do
      move input head one square to the left;
      push @   od;
   while the symbol on the stack top is @ do
      pop @;
      move input head two squares to the right  od;
   pop one symbol;
   if the symbol is 1 then move input head one square
      to the right
od
```

$|\alpha| \leq \log n$ なので，2進数 $\alpha^R \leq n$ であることに注意する．上の手続きは O(n) 時間で終了する．

命題の後半は上の手続きの逆を行う手続きを構成すればよい．これも O(n) 時間で終了する．

命題2　A を入力長の n の 2dpda とする．A の計算の途中でプッシュダウン記憶の内容が $\gamma\#\alpha, \gamma, \alpha \in \{0,1\}^*$, $|\alpha| \leq (1/2)\log n$ であったとする．すると A はプッシュダウン記憶の内容を $\gamma\#\alpha\alpha^R$ とすることができる．

証明　$\alpha = \alpha_m \alpha_{m-1} \cdots \alpha_1, \alpha_i \in \{0,1\}$ とし，@を {0,1} に属さないスタック記号とする．スタックの内容が $\gamma\#\alpha$ のとき，次の A の手続きはスタックの内容を $\gamma\#\alpha\alpha^R$ に変更する：

```
   move input head to scan the 0-th symbol;
   while the top symbol of the stack is not # do
      let the current content of the stack be
         γ#α_m α_{m-1} ··· α_i, and input head scan the w_i-th
         symbol, where w_i is the binary number defined by
         w_i = α_{i-1} α_{i-2} ··· α_1 α_1 ··· α_{i-1};
      pop α_i;
      push @ and α_i;
(1)   push w_i;
      push α_i;
      comment input head scans the 0-th symbol, and stack
         contains γ#α_m ··· α_{i+1} @ α_i ··· α_1 α_1 ··· α_i;
```

(2) *pop* $w_{i+1}=\alpha_i\alpha_{i-1}\cdots\alpha_1\alpha_1\cdots\alpha_i$, and *move* input head to scan $w_{i+1}{}^R$-th symbol;
pop @;
od;
comment input head scans $w_{m+1}{}^R$-th symbol, $w_{m+1}=\alpha_m\alpha_{m-1}\cdots\alpha_1\alpha_1\cdots\alpha_m$, and stack contains γ#;
(3) *push* w_{m+1}

push 文(1), (3) は命題1の後半を利用する. bin(j) を j の2進表示とすると, A の入力ヘッドが j 番目にあるならこれらの push 文は $bin(j)^R$ をスタックに積む. 逆に pop 文(2)は命題1の前半を利用する. $|\alpha| \leq (1/2)\log n$ なので, どの i ($1\leq i\leq m+1$) に対しても $|w_i| \leq \log n$ であり, 命題1が正しく適用できる. 上の手続きを実行することにより A のプッシュダウン記憶の内容が$\gamma\#\alpha\alpha^R$になることは明らかである.

定理1 ある定数 c>0 に対して $NSPACE(c\log n) \subseteq 2NPDA_{poly}$.

証明 M を ((1/2)log n)-1 領域限定非決定性チューリング機械とする。M はテープアルファベット {0,1} の1本のワークテープを持ち,計算の始めに M のワークテープのすべてのます目には 0 がかかれており,ヘッドは左端のます目の上にあると仮定する. Σを M の入力アルファベットとする. 定理を証明するには M のシミュレーションを多項式時間で行う 2npda A を構成すればよい. $\sigma\in\Sigma$, d∈{L,S,R} とする. M の各計算ステップに対し合成記号 (σ,d) は そのステップでスキャンしている入力記号 σ と入力テープのヘッドの動く方向 d を表す. M が m 番目のステップで入力を受理するなら m ステップ目の (σ_m,d_m) の d_m は特に意味を持たない. 都合上 d_m=S とする. 列 $\alpha\uparrow\beta$, $\alpha\beta\in\{0,1\}^*$, $|\alpha\beta| = ((1/2)\log n)-1$ は M のワークテープの内容が$\alpha\beta$で, ヘッドの位置がβの左端の記号の上にあることを意味する. $\alpha\uparrow\beta$を 0,1 のスタック記号で表すために $w\in\{0,1,\uparrow\}^*$ の符号化 [w] を考え, 0,1,↑の各記号を2ビットで表すことにする. すなわち $[\alpha\uparrow\beta]\in\{0,1\}^*$, $|[\alpha\uparrow\beta]| = \log n$.

さて A のスタック記号は 0,1 と $\{(\sigma,d) \mid \sigma\in\Sigma, d\in\{L,S,R\}\}$ を含むとする. A は次の STACKING PHASE と EXAMINING PHASE を実行することにより M をシミュレートする.

STACKING PHASE このフェーズで A は列 $(\sigma_0,d_0)(\sigma_1,d_1)\cdots(\sigma_m,d_m)$ と記号列 $[\alpha_m\uparrow\beta_m]$ を push する. ただし$\alpha_m\uparrow\beta_m$ は M の m番目のステップにおけるワークテープの内容とヘッドの位置である. A は M の状態を有限制御部に記憶している. このフェーズでの手続きを以下に示す.

guess d_0, the direction of the input head's move of M
at the initial step of M, and keep d_0 in the finite state;

(1) *push* $(\sigma_0,d_0)[\uparrow 0^{((1/2)\log n)-1}]$;

while M does not enter an accepting state *do*
move input head to scan the 0-th symbol;
let the stack contains
$(\sigma_0,d_0)(\sigma_1,d_1)\cdots(\sigma_{i-1},d_{i-1})[\alpha_{i-1}\uparrow\beta_{i-1}]$;

(2) *pop* $[\uparrow\beta_{i-1}]$, holding the value $[\uparrow\beta_{i-1}]^R$ by the input head position;

(3) *guess* the next move of M, which includes the direction d_{i-1} of input head's move;
let $\alpha_i\uparrow\beta_i$ be the content of the work tape and its head position of M at the next step;

(4) *guess* σ_i, the input symbol of M to be scaned at the i-th step;
move input head position to hold the value $[\uparrow\beta_i]^R$;

(5) *pop* $[\alpha_{i-1}]$ and *move* input head position to hold $[\alpha_i\uparrow\beta_i]^R$;
guess d_i, the input head's move of M at the step after next, and keep d_i in the finite state;

(6) *push* (σ_i,d_i);

(7) *push* $[\alpha_i\uparrow\beta_i]$

od

(1)の中の記号 $[\uparrow 0^{((1/2)\log n)-1}]$ は M のワークテープの初期内容とヘッドの位置を示す．(2)の pop 文は命題1の前半の手続きを使用する．(3)では A は M の次動作を推測するが，このうち d_{i-1} は A の有限制御部にすでに記憶してある．A は(4)で入力記号 σ_i を推測し，(6)でスタックにしまうが，このとき σ_i が "本物の" 記号かどうかは調べない．pop 文(5)は $[\alpha_i\uparrow\beta_i]^R$ を保持するため，命題1の前半の手続きを少し変更したものを使用する．push 文(7)は命題1の後半の手続きを使う．

M は多項式時間で動作するので，このフェーズで A は多項式時間で動く．

EXAMINING PHASE $[\alpha\uparrow\beta]$ の形をした記号列をスタックから取り出したのち，A は記号 (σ,d) をスタックから取り出し，推測した入力が正しかったかどうかを調べる．このフェーズの手続きを以下に示す．

```
pop [α↑β];
move input head to scan the 0-th symbol;
while there is a symbol on the stack do
    pop (σ,d);
    examine whether σ is correct, by moving the input
        head to direction d and testing that σ is the
        symbol under the input head.  If σ is not the
        symbol then halt without accepting
od;
accept
```

このフェーズが多項式時間で実行されるのは明らかである．また 2個のフェーズを実行することにより A は M をシミュレートし，A が多項式時間で入力を受理する必要十分条件は M が入力を受理することがいえる．

定理2　(1) 任意の定数 c>0 に対し NSPACE(c log n) $\not\subseteq$ $2DPDA_{poly}$ なら，$2DPDA_{poly} \neq 2NPDA_{poly}$ である．
(2) 定数 c>0 が存在して NSPACE(c log n) $\subseteq$ $2DPDA_{poly}$ なら NL $\subseteq$ LOGDCFL である．
証明　(1) $2DPDA_{poly} = 2NPDA_{poly}$ とする．すると定理1より NSPACE(c log n) $\subseteq$ $2DPDA_{poly}$ である．
(2) パディング議論による．(パディング議論は例えば [5, p.303] をみよ．)

定理3　定数 c>0 に対し ASPACE(c log n) $\subseteq$ 2NPDA.
証明　c log n 領域限定オルタネーティングチューリング機械 M を 2npda A がシミュレートすることを示す．ただし c = 1/(4 log | Γ |),Γは M のワークテープアルファベットとする．M から M と同等の ((1/4)log n)-Q 領域限定の1本のワークテープを持ち，テープアルファベット {0,1} のオルタネーティングチューリング機械 M_1 を構成できる．ただし Q は M_1 によってのみ決まる定数とする．計算の始めに M_1 のワークテープのすべてのます目には0がかかれており，ヘッドの位置は左端のます目の上にあるものとする．
Q を M_1 の状態集合，Σを M_1 の入力アルファベットとする．M_1 の様相を $C = \alpha(q,\sigma,d,k)\beta$ で表す．ただし $\alpha\beta \in \{0,1\}^*$ はワークテープの内容，$|\alpha\beta| \leq ((1/4)\log n)-Q$，ワークテープのヘッドは$\beta$の左端の記号上にある，q は現在の状態，σ はスキャンしている入力記号，$d \in \{L,S,R\}$ は次の様相への入力テープのヘッドの動く方向，q が全称状態なら k は次動作の選択番号，q が存在状態なら k=0 とする．q が全称状態なら d は C から k番目の選択動作における M_1 の入力ヘッドの動く方向であるこ

とに注意する．受理様相 $C_a=\alpha_a(q_a,\sigma_a,d_a,k_a)\beta_a$ に対し d_a,k_a は意味を持たない．都合上 $d_a=S$, $k_a=0$ としておく．一般性を失うことなしにすべての様相から r個の動作が存在すると仮定する．したがって $0\leq k<r$．さて

$$Q=3+\log\lceil|Q|\times|\Sigma|\times|\{L,S,R\}|\times r\rceil$$

としよう．すると C は ((1/4)log n)-1 記号で表される．

シミュレーションでは，A は M_1 の様相を 0,1 を使って表し，区切り記号 # と共にスタックにしまっていく．このため記号列 w の符号化 [w] を考え，[0]=00, [1]=01, [(]=[)]=10, [#]=11 とする．任意の様相 C に対して，$[C\#]\in\{0,1\}^*$, $|[0,1]|\leq$

(1/2)log n であることに注意する．A は d_0 を推測する．d_0 は M_1 の最初のステップにおける入力ヘッドの動く方向である．A は はじめに $[C_0]$ をスタックに積む．ただし $C_0=(q_0,\sigma_0,d_0,0)0^{((1/4)\log n)-Q}$ は初期様相であり，入力ヘッドは0番目の記号をスキャンしている．

A は M_1 を STACKING PHASE と EXAMINING PHASE を繰り返し実行することによってシミュレートする．これらの2個のフェーズは定理1の証明にあるものと基本的には同じである．M_1 のノーマルな計算では，繰り返しの回数は M_1 の計算木の受理様相の数だけ行われる．

STACKING PHASE　このフェーズで A は次の様相をスタックにしまう．入力ヘッドが 0番目の記号をスキャンしていると仮定する．このフェーズの手続きを以下に示す．

while the configuration $C=\alpha(q,\sigma,d,k)\beta$ (or C^R) on the stack top is not accepting *do*
move input head to scan the 0-th symbol;
(1) *pop* $[\#C]$ (or $[\#C^R]$), holding the value $[C]^R$ (or $[C^R]^R$) by input head position;
guess the k-th choice of M_1 from C for the next step if q is universal, and *guess* the next step of M_1 from C if q is existential;
let q' be the state, $\alpha'\beta'$ the content of the work tape with the head on the leftmost symbol of β' in the next step;
(2) *guess* σ', the input symbol of M_1 to be scanned in the next step;
guess d', the direction of the input head's move of the 0-th choice at the next step;

(3) *push* [C#C'^{R}] (or [C^R#C'], respectively), where
C' = α'(q',σ',d',0)β';
od

pop 文(1)は命題1の前半の手続きを使う．(2)では A は σ' を推測するだけで(3)でスタックにしまう．このとき σ' が"本当の"入力記号かどうかはチェックしない．push 文(3)は命題2の証明で与えられる手続きを少し変更したものを使う．変更したその手続きを実行することによって，A は (q,σ,d,k) を読み込み，C 中の (q,σ,d,k) の付近の記号を変えることができる．|[C#C'^{R}]| = |[C^R#C']| < log n であることに注意する．

EXAMINING PHASE ここでは A はスタックから様相を取り出し，A が推測した入力記号が正しかったかどうかを入力ヘッドの位置を移動しながら確かめていく．このフェーズの手続きは以下のようになる：

comment configuration on the stack top is accepting and the input head is on the leftmost symbol;
(1) *while* there is a configuration $C = \alpha(q,\sigma,d,k)\beta$ (or C^R) on the stack and it is either existential, or universal with $k=r$ *do*
pop [#C] (or [#C^R]);
examine whether σ is correct, by moving the input head to direction *d* and testing that σ is the symbol under the input head. If σ is not the symbol then halt without accepting
od;
if the stack is empty *then accept*;
comment C is a universal configuration, and there is the (k+1)st next move from C;
guess d', the direction of the input head's move at the (k+1)st next move from C;
(2) *pop* [#C] (or [#C^R]) and *push* [#C'] (or [#C'^{R}], respectively) instead, where $C' = \alpha(q,\sigma,d',k+1)\beta$;

(1)では C の (q,σ,d,k) を pop する．これは命題1の手続きを使用する．pop 文(2)は命題2の証明にある手続きと同様にする．

2個のフェーズを繰り返すことにより，A は M_1 の計算を正しくシミュレートし，M_1 が入力を受理するための必要十分条件は A が入力を受理することであることがいえる．

定理4 (1) 任意の定数 c>0 に対し ASPACE(c log n) $\not\subseteq$ $2NPDA_{poly}$ なら $2NPDA \neq 2NPDA_{poly}$.

(2) ある定数 c>0 が存在して ASPACE(c log n) $\subseteq$ $2NPDA_{poly}$ なら P=LOGCFL.

証明 (1) $2NPDA_{poly} = 2DPDA$ とする. 定理3より $ASPACE(c \log n) \subseteq 2DPDA_{poly}$ である.

(2) パディング議論による.

参考文献

[1] Aho, Hopcroft and Ullman, Time and tape complexity of pushdown automaton languages, Inform. Contr. 13, 186-206 (1968)

[2] Cook, Characterizations of pushdown machines in terms of time-bounded computers, JACM 18, 4-18 (1971)

[3] Galil, Some open problems in the theory of computations as questions about two-way deterministic pushdown automaton languages, MST 10, 211-228 (1977)

[4] Gray, Harrison and Ibarra, Two-way pushdown automata, Inform. Contr. 11, 30-70 (1967)

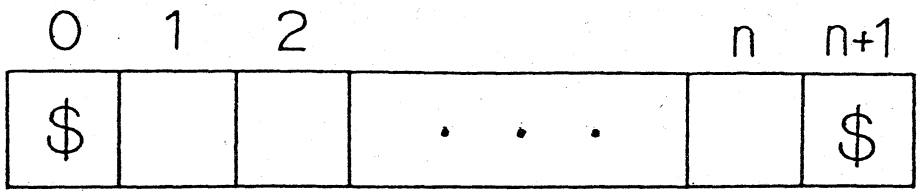
[5] Hopcroft and Ullman, Introduction to Automata Theory, Languages and Computation, Addison Wesley (1979)

[6] Ruzzo, Bounded alternation, JCSS 21, 218-235 (1980)

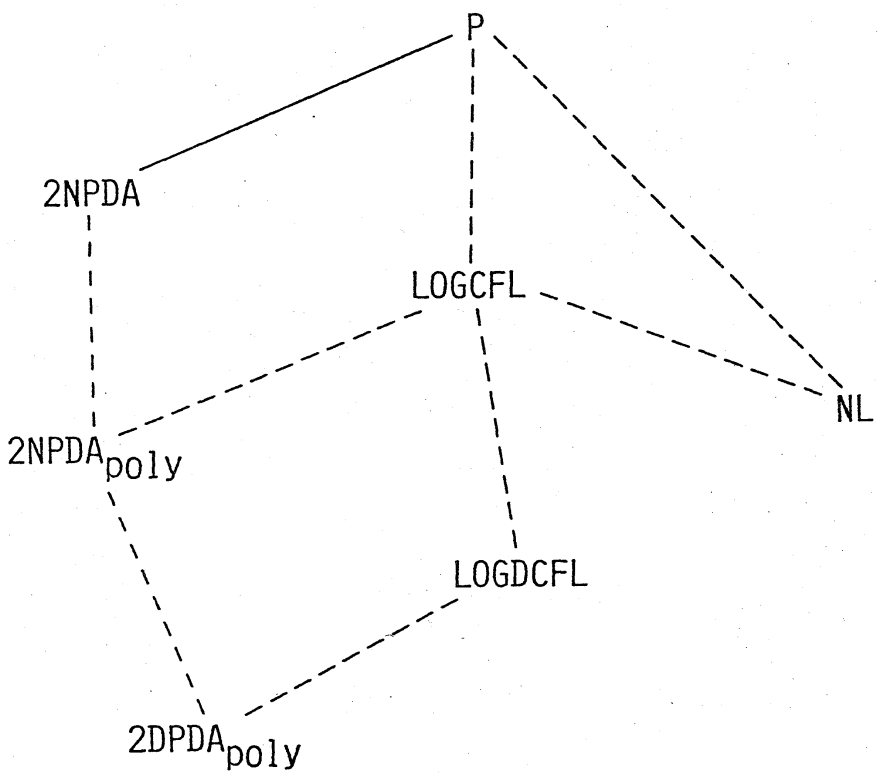
[7] Sudborough, On tape-bounded complexity classes and multihead finite automata, JCSS 10, 62-76 (1975)

[8] Sudborough, A note on tape-bounded compexity classes and linear context free languages, JACM 22, 499-500 (1975)

[9] Sudborough, On the tape complexity of deterministic context free languages, JACM 25, 405-414 (1978)



第1図　　入力テープ



第2図　　複雑さの包含関係